Integrated Architectural Level Power-Performance Modeling Toolkit

David Brooks
Division of Engineering and Applied Sciences
Harvard University

In recent years power has joined performance as a first-class design constraint for nearly all types of computer systems -- from low-end embedded microprocessors intended for PDAs to server-class microprocessors for multi-way SMP systems. While power-aware research has been conducted at all of these levels, studies have frequently been compartmentalized into one particular design point. This compartmentalization has in part been forced due to a lack of infrastructure for exploring ideas across a wide range of systems.

Studies done at the architectural level frequently rely on processor performance simulators written in high-level languages such as C, rather than VHDL-level processor descriptions, because of the advantages in modeling time, model flexibility, and simulation speed. To be useful in the early stage of the architectural definition process, our power-performance simulation infrastructure must be integrated with these high-level architectural models. In recent years, there has been an increasing interest in the integration of high-level performance simulators with power models. A significant shortcoming of all existing power-performance evaluation systems is that they are often intended for one particular design points. For example, Wattch [1] and SimplePower [2] were primarily intended for the high-performance and embedded computing spaces respectively. No infrastructure currently exists that scales power models from the embedded to the high-end design space.

We are currently developing a robust, integrated infrastructure for studying power-performance issues across a range of systems. By leveraging a common ISA and shared simulation infrastructure, we will be able to perform apples-to-apples comparisons between processors intended for specific design spaces. For example, recently there has been significant attention brought to the idea of reusing microprocessor cores in multiple design spaces. In particular, there has been much interest in exploring the possibility of using multiple low-power, embedded processors in blade systems or SMP-on-a-chip designs for server workloads. There has also been interest in taking server-class microprocessors and bringing them into use in lower-end systems. For example, the processor core of the original POWER4 microprocessor has recently been introduced as the PowerPC970 -- a 64-bit microprocessor for use in blade servers and desktop (and potentially laptop) systems.

We utilize the MET/Turandot toolkit originally developed at IBM TJ Watson Research Center as the underlying PowerPC microarchitecture performance simulator [3]. Turandot is flexible enough to model a broad range of microarchitectures and has undergone extensive validation [3]. In addition, Turandot has been augmented with power models to explore power-performance tradeoffs in an internal IBM tool called PowerTimer [4]. Turandot is freely available to the research community through licensing arrangements with IBM, and we are currently working with IBM to develop an external, public release of PowerTimer.

Figures 1 and 2 are examples of the type of results that we plan to collect with this toolkit. We have used our simulator to model the characteristics of three machine organizations – "*highperf*" (an aggressive, very-wide issue, out-of-order superscalar machine appropriate for workstations and servers), "*midperf*" (a high-performance embedded superscalar microprocessor appropriate for network processors and other high-end embedded applications), and "*lowperf*" (a single-issue embedded microprocessor intended for PDAs and very low-power applications). In Figure 1, we show the Instruction per Cycles (IPC) estimates for the three machines across a range of embedded (MiBench) and workstation (SPEC2K) applications. Figure 2 shows the relative power-performance efficiency of the machines in MIPS per Watt (energy), $MIPS^2$ per Watt (energy-delay product), and $MIPS^3$ per Watt (frequency/voltage scaling invariant metric). Interestingly, the highest performance microprocessor produces the best $MIPS^3$ per Watt score while the two embedded processors perform much better on the less performance-centric

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
20 AUG 2004	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Integrated Architectural Level Power-Performance Modeling Toolkit	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Division of Engineering and Applied Sciences Harvard University**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**13**	

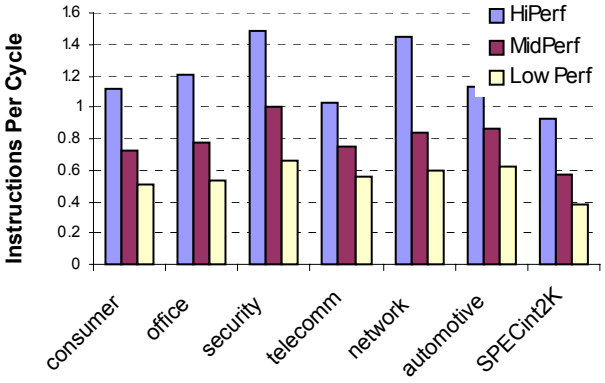


Figure 1. IPCs for Embedded and Workstation Apps

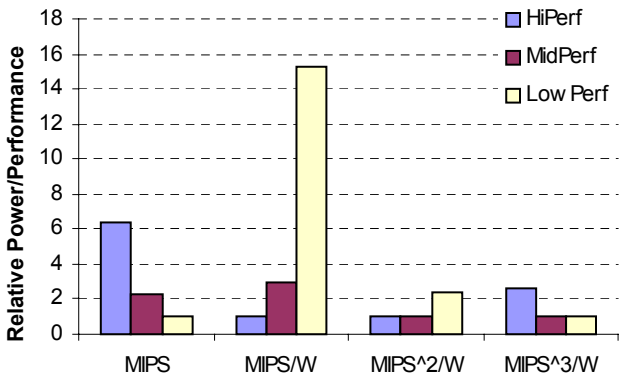


Figure 2. Relative Power-Performance

metrics. In this very simple study, we have assumed numbers representative of the order of magnitude of power dissipation for these classes of microprocessors; 0.5W for *lowperf*, 5W for *midperf*, and 50W for *highperf*. Future studies will use power models under development to perform more detailed analysis.

The major thrust of our current work is to extend the scalability and flexibility of the power-performance model and beginning to study the power-performance characteristics of multiple design points. Existing work has focused on scaling PowerTimer's power models for resource sizes [4] and pipeline depth [5]. Our current efforts seek to maintain and improve the existing scalability models while also developing models for the following:

Pipeline Width – Similar to the pipeline depth, the fetch, dispatch, and retire width of the machine has a significant impact on the power-performance efficiency of the machine. Since pipeline width varies considerably between embedded and high-performance microarchitectures, this will be a key parameter that we will explore when studying the power-performance efficiency of design spaces.

Multithreading and Chip Multiprocessing – Both simultaneous and coarse-grained multithreading have emerged as key microarchitectural techniques and detailed study of the power-performance efficiency of these schemes in the context of the other architectural parameters will be enlightening. In addition, the recent trend to multiple on-chip processor cores provides a rich area for power-performance tradeoffs for systems with sufficient thread-level parallelism.

Circuit Hardware Intensity – The hardware intensity metric describes the relative circuit delay vs. circuit power dissipation through device size tuning and logic restructuring [6]. Hardware intensity, along with architectural decisions, can have a key impact on the overall power-performance efficiency of the design. Tradeoffs between circuit hardware intensity and architectural power-performance choices will be a key effect that we will study. Tuning the circuit hardware intensity is a key method to scale our models from embedded, low-power processors to high-performance microarchitectures.

The development of an integrated, scalable power-performance modeling toolkit will allow design space studies across a wide range of modern machine architectures. These studies will help designers answer questions about design tradeoffs within a particular microarchitecture as well as the power-performance efficiency of applying an existing core design to alternate design spaces.

[1] D. Brooks, V. Tiwari, and M. Martonosi. "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *27th International Symposium on Computer Architecture (ISCA)*, June 2000.

[2] N. Vijaykrishnan, M. Kandemir, M.J. Irwin, H.S. Kim, W. Ye, "Energy-driven Integrated Hardware-Software Optimizations using SimplePower," *27th International Symposium on Computer Architecture (ISCA)*, June 2000.

[3] M. Moudgill, J. Wellman, and J. Moreno. "Environment for PowerPC Microarchitecture Exploration," *IEEE Micro*, 13(3):9-14, May/June 1999.

[4] D. Brooks, P. Bose, V. Srinivasan, M. Gschwind, P. Emma, M. Rosenfield, "Microarchitectural-Level Power-Performance Analysis: The PowerTimer Approach," *IBM Journal of Research and Development*, Oct/Nov 2003.

[5] V. Srinivasan, D. Brooks, M. Gschwind, P. Bose, V. Zyuban, P. Strenski, and P. Emma, "Optimizing Pipelines for Power and Performance," *35th International Symposium on Microarchitecture (MICRO-35)*, November, 2002.

[6] V. Zyuban, P. Strenski, "Unified Methodology for Resolving Power-Performance Tradeoffs at the Microarchitectural and Circuit Levels," International Symposium on Low-Power Electronics and Design, Aug 2002.

Why Model Power at the Architectural Level?

- Modeling power at the architectural level allows tradeoffs between HW and SW
- Architectural decisions substantially impact both performance and power
- Architectural decisions often cannot be reversed – must estimate power early

Power-Performance Modeling Infrastructures

Analytical Tools:
CACTI, Wattch, PowerAnalyzer

Mixed-Mode Tools:
Tempest, Accupower

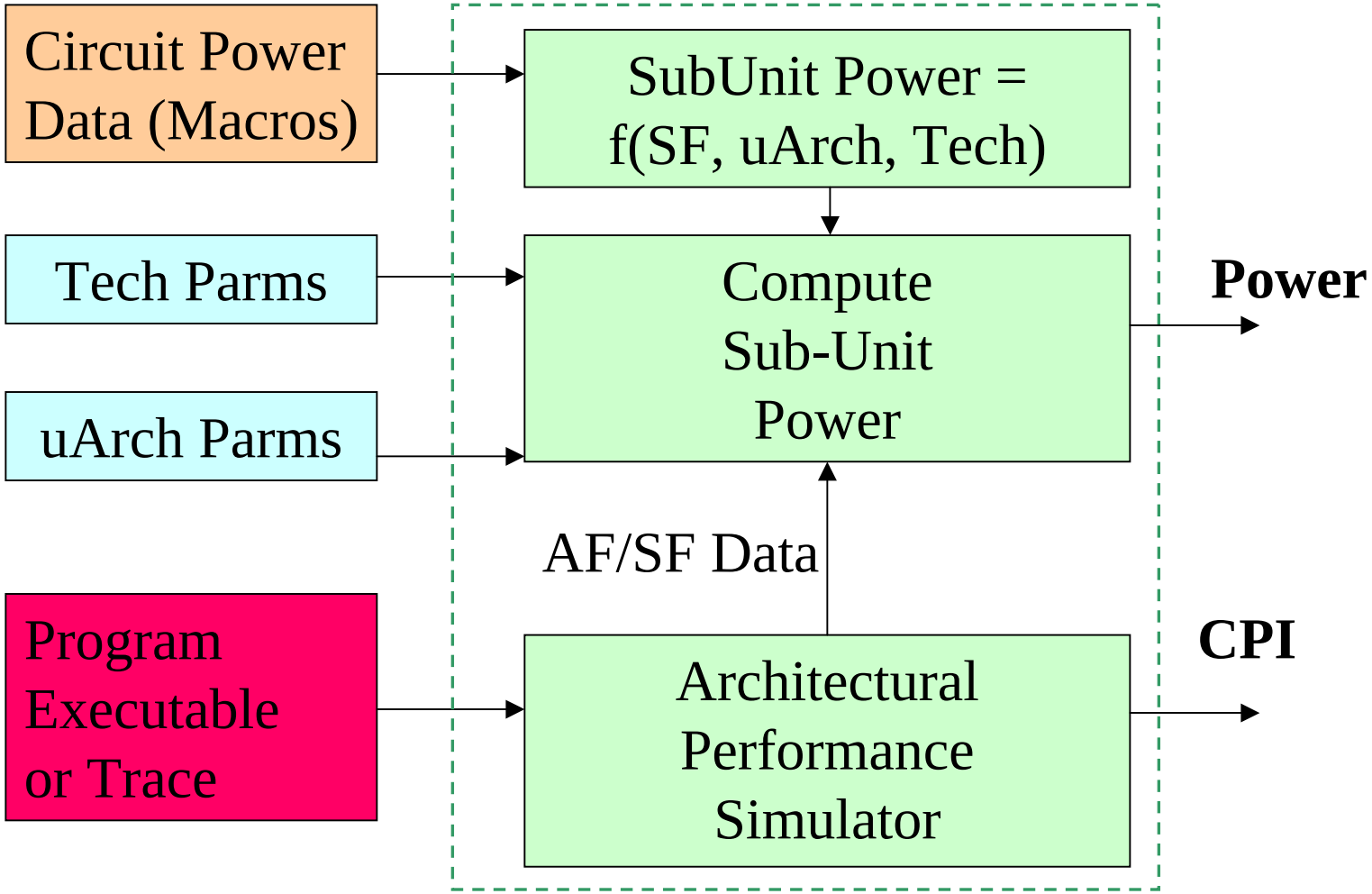
Empirical Tools:
SimplePower, PowerTimer

Increased Flexibility

Ease of Validation

- Existing tools require tradeoffs between flexibility and ease of validation/accuracy

PowerTimer

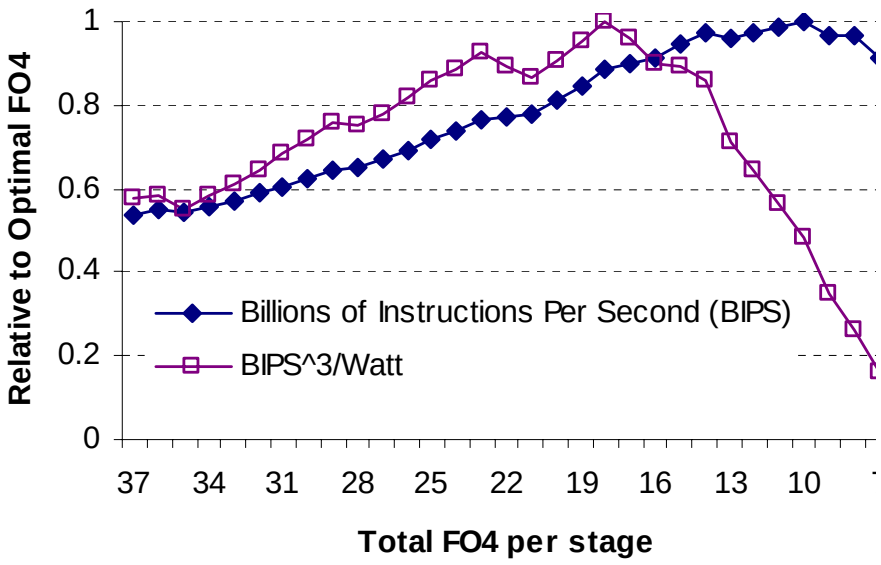

Circuit Power Data (Macros)
SubUnit Power = f(SF, uArch, Tech)
Tech Parms
Compute Sub-Unit Power
Power
uArch Parms
AF/SF Data
Program Executable or Trace
Architectural Performance Simulator
CPI

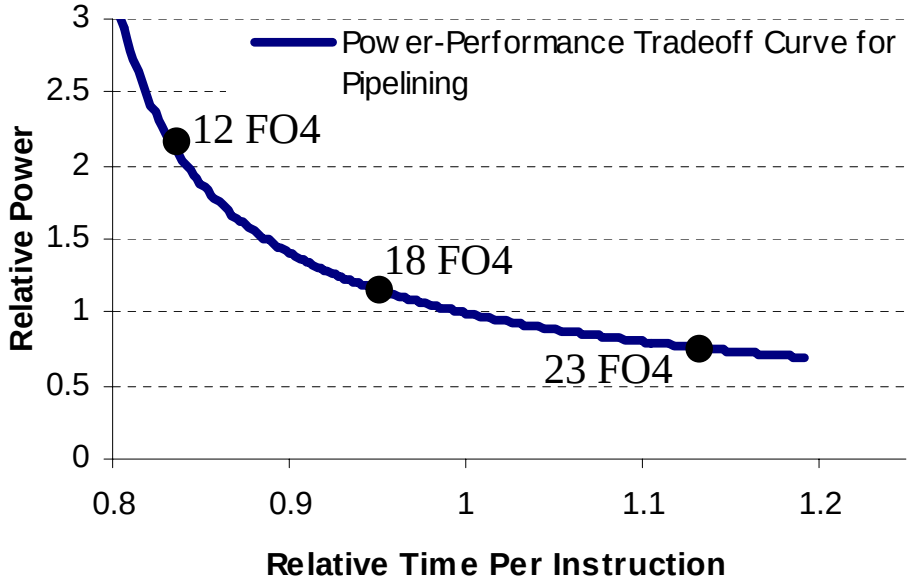
PowerTimer Observations

- PowerTimer works well for POWER4 and derivatives
  - Scales well from base microarchitecture
  - Lack of detailed (bit-level) SF not seen as a problem for high-performance chips (seen as noise)
    - Chip level SFs are quite low (5-15%)
    - Most (60-70%) power is dissipated while maintaining state (arrays, latches, clocks)
    - Much state is not high-level anyway (available in early-stage timers)
  - Validation -- Based on validation of individual pieces
    - We know how to validate the performance model (more or less)
    - Power estimates from circuits are accurate
    - Circuit designers vouch for clock gating scenarios

How are these tools used?

- Architecture and software analysis and tradeoff studies, e.g. optimal pipeline depth



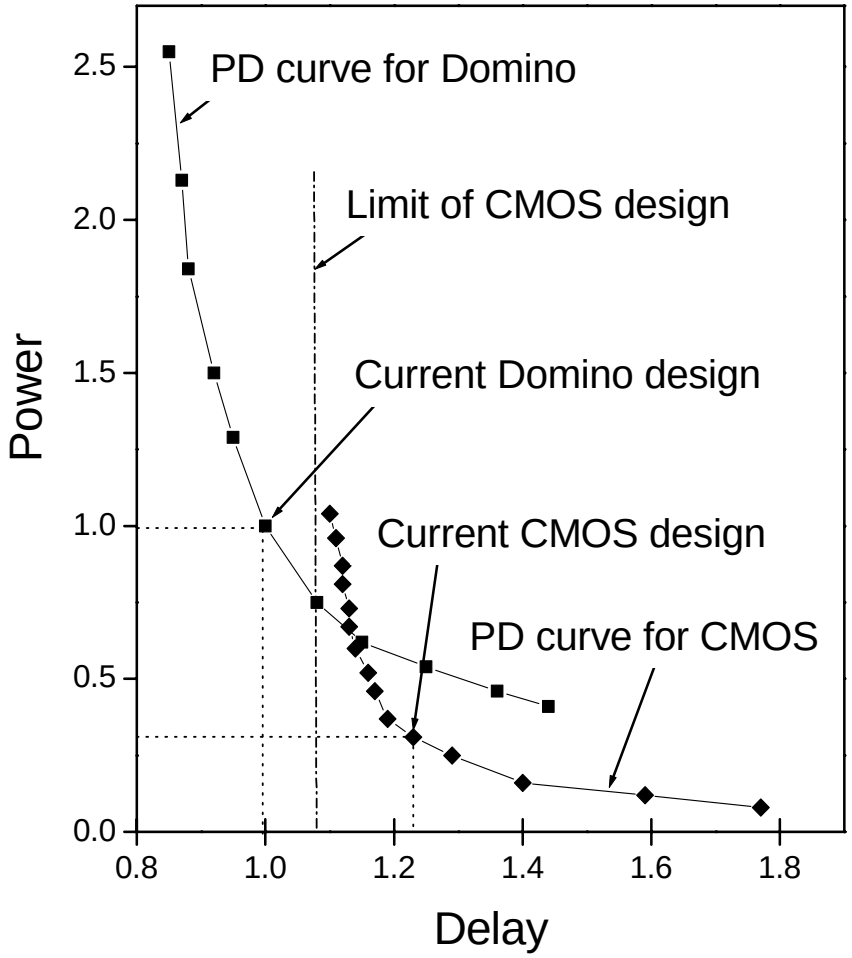


Optimal Pipeline Depth for Microprocessors, from MICRO2002

What are current tools missing?

- Models should account for circuit and architectural level power-performance tradeoffs
- Increased flexibility, accuracy, and ease of validation
- Integrated models for delay, power, and design complexity

Circuit-level Power-Performance Tradeoffs

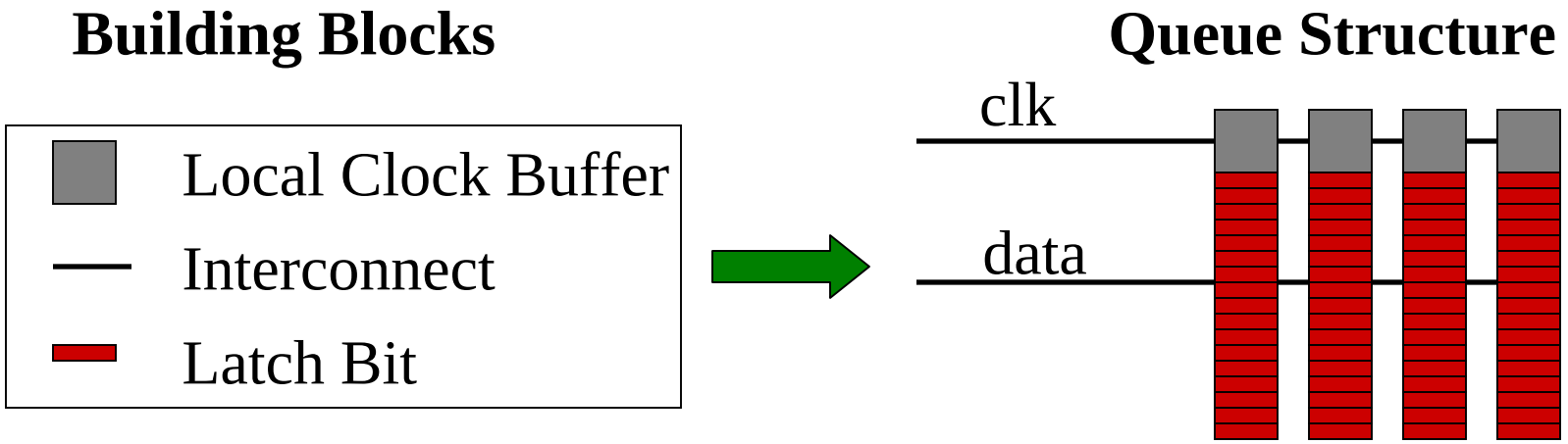


- Circuit/architecture decisions should be made together
- Allows joint-optimization of the power-delay curves

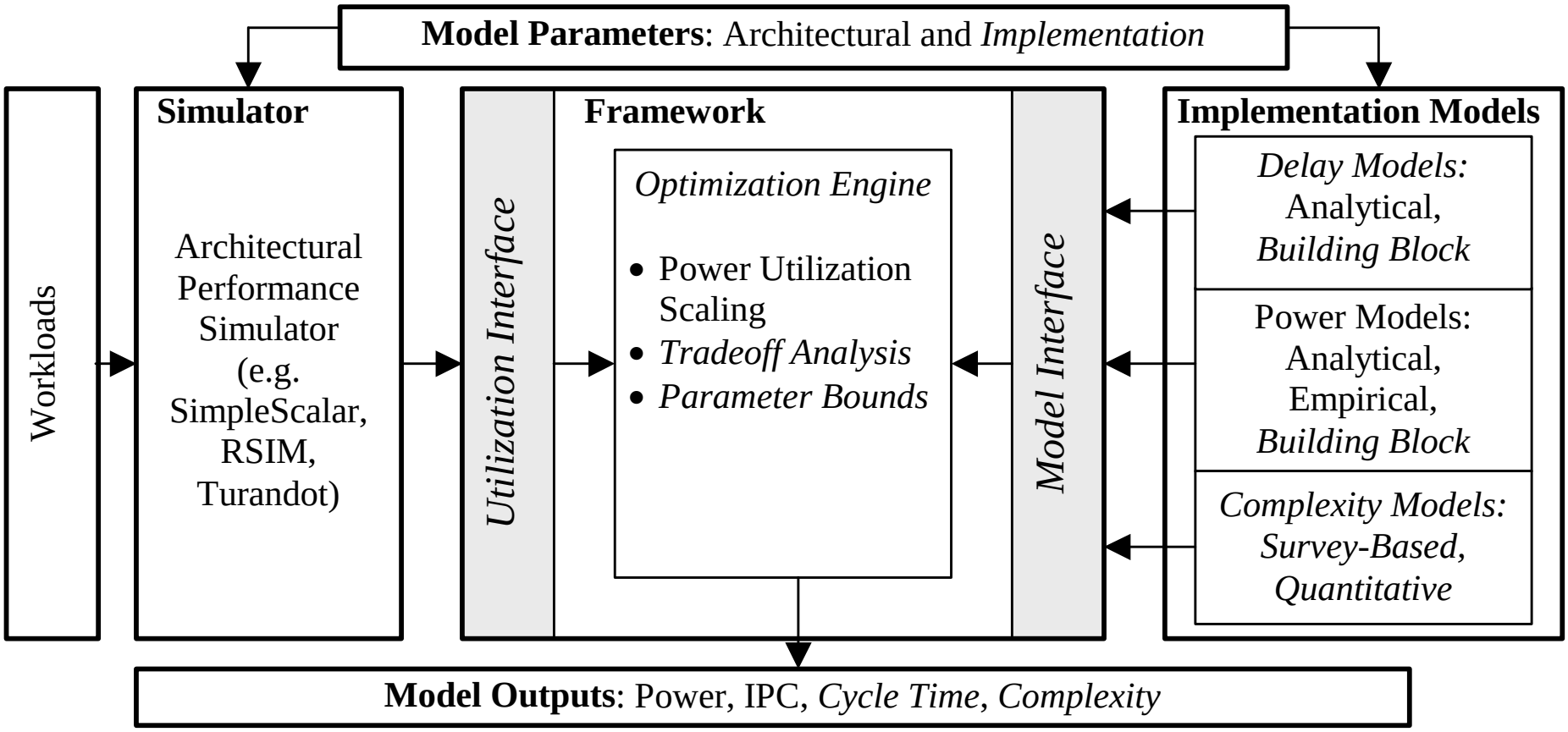
32-bit Adder Power/Delay varying Circuit Style and Transistor Sizings
From Tiwari, et. al. DAC98

Implementation Models

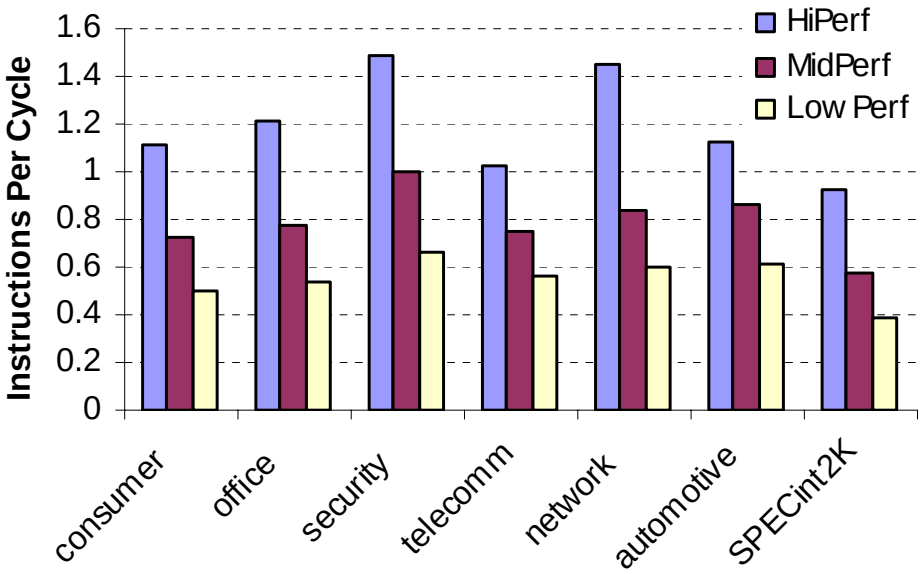
- Building block methodology can provide flexibility with ease of validation
- Develop/Validate models for common intrinsic blocks (latch, mux, interconnect, etc)
  - Chain building blocks together to model higher level structures (issue queue, register file)

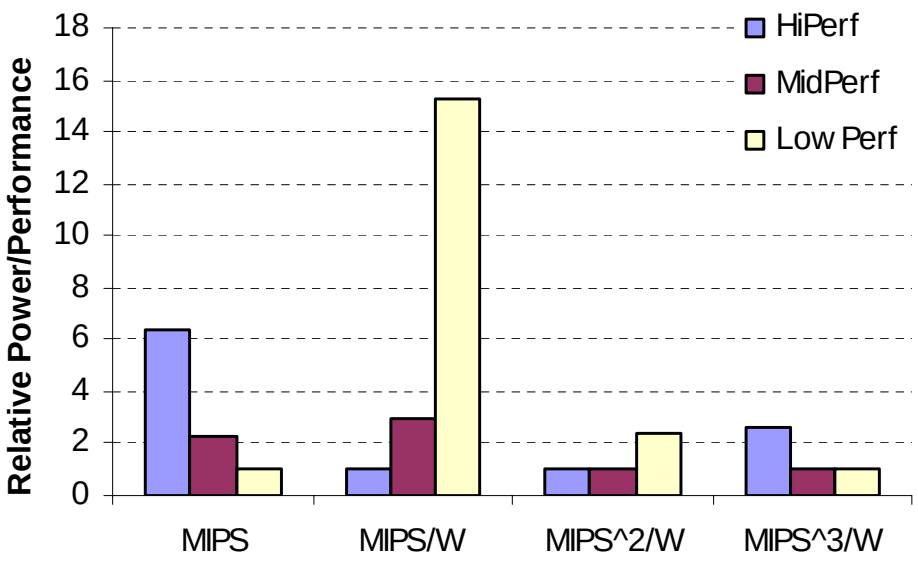


Model Framework



Example of Experiments





- Consider three architectures of varying complexity
- Which one is “optimal” for power-performance efficiency?
- What about design points between these choices?

Future Work

- Scalable models for power/performance allow seamless analysis of high-performance embedded architectures
- Development of design complexity models
- Optimization infrastructure to study joint tradeoffs